

# Hyper-navigation in Virtual Buildings

Jürgen Emhardt, Jürgen Semmler, Thomas Strothotte  
Freie Universität Berlin  
Department of Computer Science  
Takustraße 9, 14195 Berlin, Germany  
e-mail: emhardt@inf.fu-berlin.de

## *Abstract*

The Virtual Reality (VR) user interface style allows the user to manipulate virtual objects in a 3D environment. We introduce a software architecture which guides users in virtual worlds. The architecture is based on the concept of task-oriented agents which support users in acting within a Virtual Environment (VE). These agents use an application independent toolbox, the hyper-renderer, to retrieve information that is calculated during the rendering process. We have used a virtual building to implement various prototypes in order to evaluate our approach.

Keywords: task-oriented agents, navigation, hyper-navigation, orientation, exploration, hyper-rendering.

## 1. Introduction

During the last years Virtual Environments (VEs) have found more and more attention. Emphasis has been laid on real-time animations, 3D input devices and the mapping between I/O and VR. Quite a lot of different input devices have been used in an attempt to give the user the freedom of 3D interaction. For the most part only the technical and algorithmic aspects of navigation have been discussed (for example Pletincks [8]). This means that applications map the user's input into a VE and let the user move through the virtual world. It does neither mean that users are supported in gathering information about the world for exploration, nor that users are assisted if they get lost in a large VE.

We believe that these facts are the major drawback of existing VR applications. The user who uses VR equipment cannot escape from the virtual world. This paper therefore presents a new architecture to guide users within a VE and support them with additional information about their environment. Task-oriented agents can be triggered by the system as well as by the user to support particular actions. These agents use information that is retrieved from a database built using the hyper-renderer which extracts semantic relations within a VE during the rendering process. We introduce the concept of *hyper-navigation* which allows users not only to navigate, but also to explore their environment with respect to spatial, temporal, and semantic relationships between objects. Furthermore, users can use the VE as a testbed for experiments. For example, a room in a virtual building can be used to investigate the spatial impression of an overhead transparency which has been produced by using a usual graphics editor. A presentation agent can determine the

good and poor legibility regions within this room. Finally, the concept of hyper-navigation covers the use of virtual buildings and their interior as a metaphor. For example, Robertson et al. [9] use 3D rooms and doors as metaphors in their Information Visualizer system.

This paper is organized as follows: Related work is surveyed in Chapter 2. Chapter 3 gives detailed information about the architecture and the agents of our software system. Chapter 4 presents our prototypes and chapter 5 discusses future research.

## **2. Related Work**

Visibility preprocessing for interactive walkthroughs through complex environments has been done by Teller and Sequin [13] and Airey [2]. In addition, Airey precomputes radiosity. However, the methods applied work on a polygonal basis of the objects and do not use symbolic names. Therefore, equipment like furniture and their functionality is not taken into account.

Shaw et al. [10] developed a VR-framework which facilitates the development process of VR applications in general. Their toolkit provides support for distributed computing, head-mounted displays, room geometry, hand input devices, and sound feedback. This framework supports navigation in a technical sense.

Zeltzer et al. [15] at MIT developed a general purpose package for building interactive simulation systems, especially for task level animation systems. They use a constraint network to connect all their objects. If an object changes its state all constraints which are involved are evaluated through the network. A data glove is used to allow the user to interact with the simulation environment.

Robertson et al. [9] at Xerox Parc developed an architectural model for VR user interfaces called the Cognitive Coprocessor Architecture. This approach is based on multiple asynchronous, interactive agents and smooth animation. Their agent concept is similar to ours. They try to maximize the available information on a screen (the "Small Screen Problem"), whereas our approach is oriented towards the need to guide users of VR applications and support them with additional information.

Zelevnik et al. [14] described an object-oriented framework to integrate a variety of simulation and animation paradigms. Their framework is an attempt to expand the traditional modelling tools with 3D widgets, a time model, and interactive animation. Their object-oriented system uses a delegation hierarchy to let the involved objects change all their attributes dynamically.

## **3. System Design**

Our architecture follows the idea presented in Figure 1 and is described in detail in Emhardt [4]. The inner circle in the figure depicts pure movement and navigation whereas the outer circle stands for the ability of an agent supported exploration. The user can navigate interactively through the scene by manipulating the camera. The hyper-renderer supports agents with necessary information about the model state (Emhardt and Strothotte [5]).

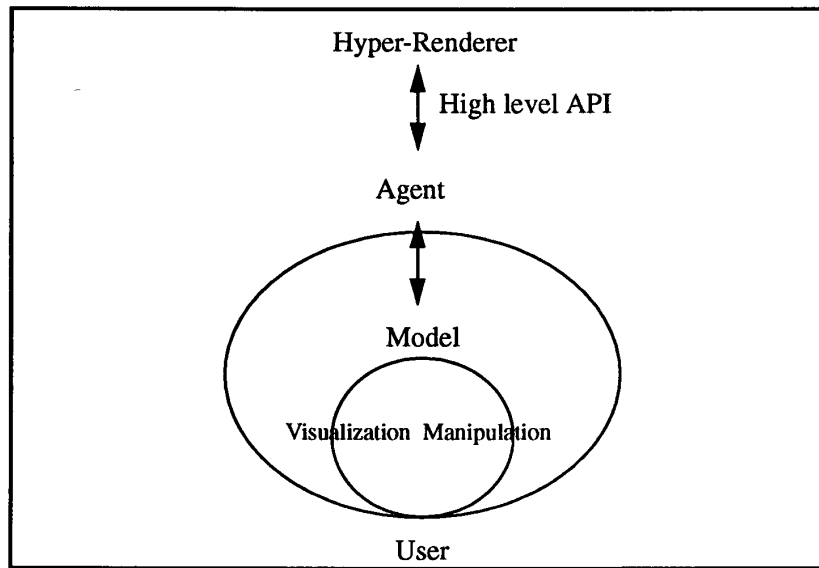


Figure 1. *A software architecture for agent-supported exploration.* The inner circle depicts the ability of navigation, whereas the outer circle enables users to explore their virtual environment by communicating with task-oriented agents.

### 3.1 Agents

The agent metaphor is widely used in the fields of artificial intelligence and user interface design. Mostly, the term refers to autonomous agents that are able to achieve goals by planning, for example (Genesereth and Nilsson [7]). Especially user interface designers use agents to take the burden of time consuming and monotonous retrieval of information from the user. Agents are used for doing this kind of work. The results are presented to the user in a form that depends on the underlying system. A standard example in this field is a mail agent that filters and sorts incoming mail and throws junk mail away.

The agent metaphor we are using has been introduced by Sufrin and He [12]. They developed a theory of communicative interactive processes, which they called agents, to design interactive systems. We follow their definition of interactive systems as a collection of communicative interactive processes representing both the computer system and its user. The entire system is built up from smaller components by describing how the parts interact with each other. The practical use in user interface design is described in Abowd [1].

### 3.2 Why Task-oriented Agents?

The Cognitive Coprocessor Architecture (Robertson et al. [9]) is based on a three-agent model: a user, a user discourse machine and an application. The application may be broken into subagents that supply services. In contrast to that we use agents in a different way. We do not try to map parts of the system onto agents but let our agents solve tasks for the user. In any domain there are various specific tasks that occur from time to time with different frequency. When users enter the VE

our agents might be triggered to investigate things other than themselves. We associate agents with graphical objects in the environment that make objects show their behavior. These agents are active and reactive. Thus objects (resp. their agents) can be involved in actions either triggered by other objects or triggered by the user.

This approach helps us to separate the user interface from the underlying algorithms. As Abowd [1] shows, it is more natural to use a high-level description which is mathematically based and free of implementation clutter than a programming language.

### **3.3 A New Concept: Hyper-Navigation**

With the concept of hyper-navigation, we intend to make VEs more intelligent and support users in performing different tasks. The main purpose of hyper-navigation is to prevent users from getting lost in large VEs (transperceptual spaces) and to support user-driven experiments with critiquing systems. Critiquing systems as introduced by Fisher et al. [6] support users in their problem solving and learning activities. The core task of critics is the recognition of deficiencies in a solution and communication of those observations to users.

In particular, hyper-navigation covers the following tasks which are supported by agents:

- 0) (Usual) Navigation, i.e. the possibility of moving from one point to another and the possibility of orientation;
- 1) Symbolic visibility preprocessing and command-oriented navigation as described in Emhardt [4];
- 2) Exploration of spatial, temporal, and semantic relationships between objects;
- 3) Using the VE as a testbed for user-driven experiments;
- 4) Using the VE as a metaphor for the visualization and understanding of abstract data and conceptual relationships.

We have built a number of prototypes to evaluate the idea of using agents to support the user in exploring and using a VE. In the following chapter, we discuss two of our applications: A user guide to explore a virtual building, and an agent which supports users to present information in virtual rooms. The first application refers to item 2) (spatial exploration), and the second prototype refers to item 3) (experiment). The agents are specified in Z [11] and use the hyper-rendering software to access information about the geometry of the model and the visibility region of the user.

## **4. Applications**

### **4.1 A User Guide for Spatial Exploration**

We implemented a tour guide which guides visitors of a virtual building to any destination inside the building. This tour guide works in two different modes: In the first mode, users can navigate independently, while the agent presents sign-posts at appropriate locations. For example, if the user tells the agent "I'd like to move to the lobby", an arrow pointing towards the proper direction will be seen during the navigation process. If the user looks into a hallway which does not lead towards his destination, a "do not enter-sign" is shown. A detailed discussion of this application including the corresponding Z schemata can be found in Emhardt [4].

In the second mode users need not to navigate independently, but animations are presented to them showing an appropriate way to their destination. In this mode, users get answers to questions like "Where am I?" by predefined macros using the API of the hyper-renderer. Furthermore, users can stop a presented animation and switch to the other mode for independent navigation (and vice versa).

## **4.2 A Presentation Agent**

We implemented a critics system for the design and investigation of overhead transparencies and slides in a 3D world. Users can prepare a presentation by using a usual graphics editor and then project the resulting overhead transparencies or slides onto a (virtual) wall inside a virtual room. They can choose between virtual rooms of different sizes and locate themselves at different seats. The illumination is done by using a radiosity algorithm. The critiquing system uses knowledge which can be derived from usual desktop publishing books like Conover [3] and works as follows: To each viewing location, the lower bound of legibility is determined by using optical rules. If the scaling factor of a text or graphics symbol within the prepared presentation is smaller than this value, the critics system presents an appropriate hint. Furthermore, the critics classifies the legibility regions and shows them in an appropriate color.

Figure 2 shows a projected overhead transparency in the lecture hall of our institute. Note that the stick figure symbolizing the user and the inner and the outer circles of our software architecture (see Figure 1) have poor legibility. The critics system shows these parts of the slide on the original page in red color. Figure 3 shows the different legibility regions inside the lecture hall for this graphics slide in appropriate colors.

## **5. Conclusion and Future Work**

We introduced the concept of hyper-navigation in order to support users in performing different tasks. Each task is supported by a task-oriented agent. The agents can be defined formally by using languages like Z and CSP. The implementation is made possible by using a new architecture which in turn makes use of an application independent toolbox, the hyper-renderer. As examples, we presented two prototypical applications.

Our future work will concentrate on integrating the results of the projects described above and on enhancing the functionality of the equipment in our virtual buildings. Agents should be able to react on and to perform dynamic changes in the environment. Furthermore, we want to use a virtual building as a metaphor for visualizing hierarchically organized data. The tour guide can then be used to guide users through abstract data sets and not only through existing buildings.

## **6. Acknowledgments**

We would like to thank Matthias Lehmann and Dirk Krause for implementing the API of the hyper-renderer. The design and the implementation of the tour guide and the presentation agent was done by Nils Kremser and Ralf Preininger in their master's theses. Martin Kurze prepared the example photographs discussed in this paper and Debora Weber-Wulff as well as Lars Reichert gave many helpful comments.

## 7. References

1. ABOWD, G.D. Agents: Communicating interactive processes. In *Proc. IFIP TC 13 Third International Conference on Human-Computer Interaction (INTERACT'90)*, Cambridge, 143-148.
2. AIREY, J.M., ROHLF, J.H., BROOKS, F.P. Towards image realism with interactive update rates in complex virtual building environments. In *Computer Graphics*, 24, 2, (1990), 41-50.
3. CONOVER, Th.E. *Graphic communication today*, West Publishing Company, 1985.
4. EMHARDT, J. User guidance in virtual worlds. In *Proc. Computer Graphics International'93 (CGI'93)*, Lausanne, Springer-Verlag, (1993).
5. EMHARDT, J., STROTHOTTE, Th. Hyper-rendering. In *Proc. Graphics Interface'92*, Vancouver, 37-43.
6. FISCHER, G., LEMKE, A.C., MASTAGLIO, Th. Using critics to empower users. In *Proc. CHI'90*, 337-347.
7. GENESERETH, M.R., NILSSON, N.J. *Logical foundations of artificial intelligence*. Morgan Kaufmann Publishers, Los Altos, 1987.
8. PLETINCKS, D. The use of quaternions for animation, modelling and rendering. In *New Trends in Computer Graphics (Proc. CGI'88)*, Springer-Verlag, (1988), 44-53.
9. ROBERTSON, G.G., CARD, S.K., MACKINLAY, J. Information visualization using 3D interactive animation. In *Communications of the ACM*, 36, 4, (1993), 56-71.
10. SHAW, Ch., LIANG, J., GREEN, M., SUN, Y. The decoupled simulation model for virtual reality systems. In *Proc. CHI'92*, 321 -334.
11. SPIVEY, J.M. *Understanding Z: A specification language and its formal semantics*, Cambridge University Press, 1988.
12. SUFRIN, B., HE, J. Specification, refinement and analysis of interactive processes. In *Formal methods in human computer interaction*, M.D. Harrison and H.W. Thimbleby, Eds. Cambridge University Press, Cambridge, (1990), 153-200.
13. TELLER, S.J., SEQUIN, C.H. Visibility preprocessing for interactive walkthroughs. In *Computer Graphics*, 25, 4, (1991), 61-69.
14. ZELESNIK, R.C., CONNER, D.B., WLOKA, M.M., ALIAGA, D.G., HUANG, N.T., HUBBARD, P.M., KNEP, B., KAUFMAN, H., HUGHES, J.F., VAN DAM, A. An object-oriented framework for the integration of interactive animation techniques. In *Computer Graphics*, 25, 4, (1991), 105-111.
15. ZELTZER, D., PIPER, S., STURMAN, D.J. An integrated graphical simulation platform. In *Proc. Graphics Interface'89*, London, Ontario, 266-274.

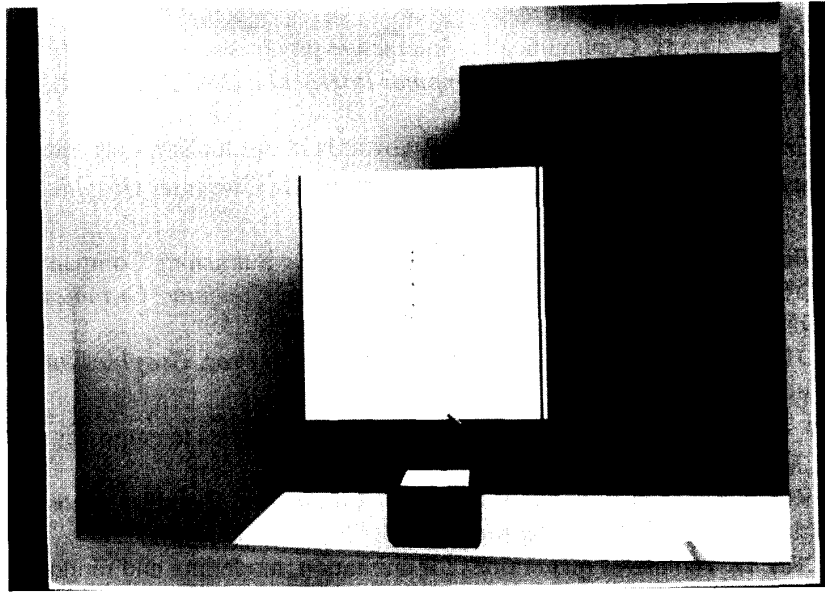


Figure 2. A projected overhead transparency in the lecture hall of our institute. The slide shows the software architecture presented in Figure 1. Note that the stick figure symbolizing the user and the inner and the outer circles have poor legibility.

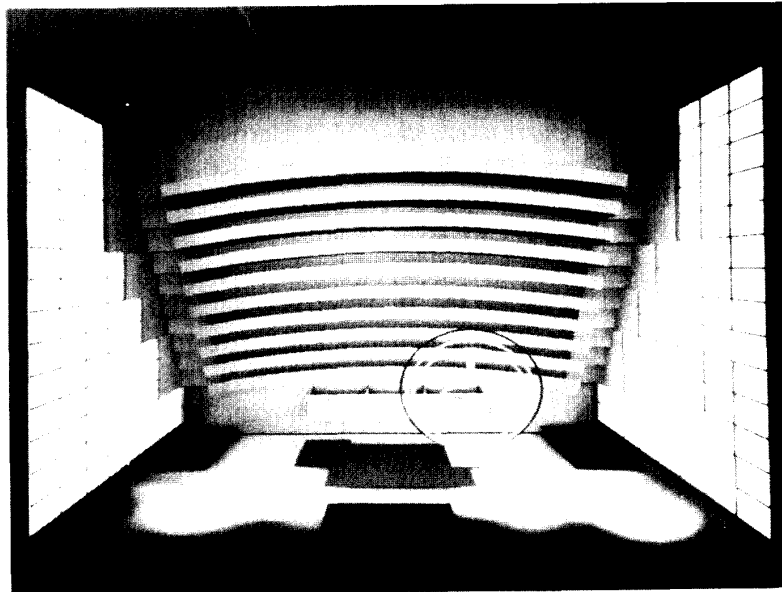


Figure 3. Legibility regions inside the lecture hall for the graphics slide of Figure 2. As some parts of the slide have poor legibility, the corresponding legibility regions which are shown in appropriate colors are relatively small.